Object Transformation Modeling

Combining
Data / Object Modeling
with
State-Transition Modeling

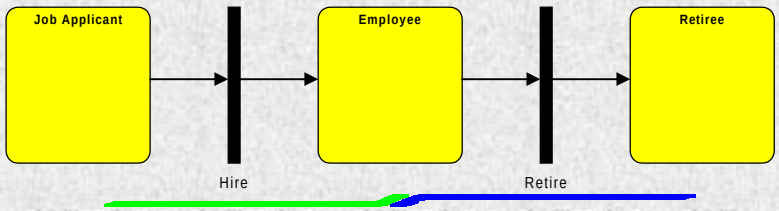
Jim Fulton
Advanced Modeling Interest Group
October 20, 1998

ERDs v STDs

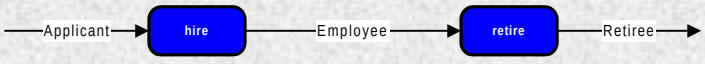
- *Entity-Relationship Diagrams / Object Models*
  define *static* rules for
  - Entities
  - Attributes
  - Relationships
  - Inheritance
- *State Transition Diagrams / Petri Nets*
  define *dynamic* rules for
  - States
  - Changes of State
  - Substates

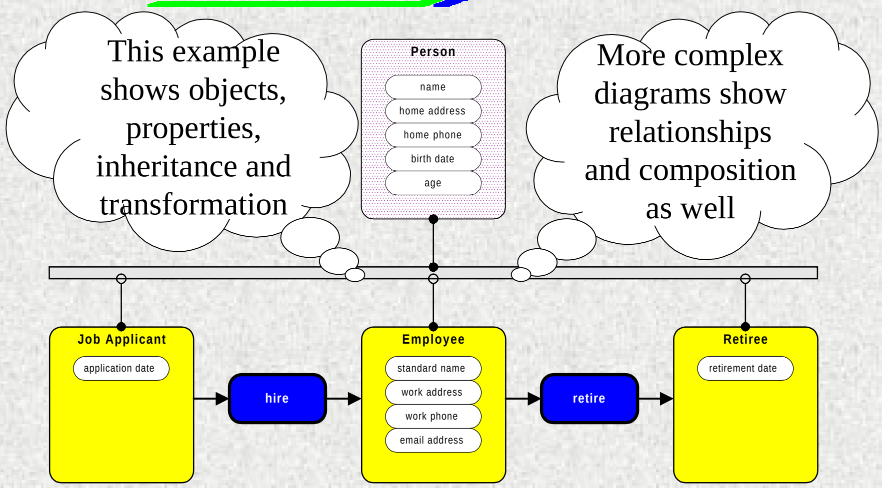
What's the connection?

Typical Example in *State-Transition Diagram* Notation



Typical Example in *Data Flow Diagram* Notation





October 20, 1998 Object Transformation Modeling 5

States as *Object Classes*? As *Tables*?

- Object Instances Migrate among Classes!
- Not Supported by OO Theory
  (Not Excluded Either!)
- Not Supported by Relational Theory
  (Not Excluded Either!)
- Not Supported by OO or Relational Tools
  (Requires Intelligent Application of Tools)

October 20, 1998 Object Transformation Modeling 6

What Object Transformation Adds to ERDs

- Dynamic as well as Static Rules
- Explicit Association of Data / Object Types with Transformation Rules that Specify Code
- Validation of Inheritance Hierarchy

What Object Transformation Adds to STDs

- Static as well as Dynamic Rules
- Explicit Association of Transformation Rules with Data / Object Types
- Inheritance Hierarchy as Basis for Transformation
- ***Object Model as Roadmap to Functionality***

Benefits of Object Transformation Modeling

- Simplified Roadmap to States and Functions
  - Makes inheritance visible in state specification
  - Makes states reusable objects
  - Enables traceability between state-methods and object model
- Integrated, Declarative Specification of both Static and Dynamic Rules
  - Bases transformations on well-defined objects
  - Avoids algorithmic control structures in specification
    - Case structures specified through subtyping
    - Iterations specified through collection objects
    - Sequence specified through input dependencies
  - Simplifies user validation
- Potential Code Simplification
  - Allows control structures in state-specific code to be replaced by encapsulated, polymorphic methods

Object Transformation Availability

- No current support by graphic CASE tools
- High level of support in EXPRESS-2 standard
  - Unclear what support vendors will provide
  - SDAI offers standard meta-model for details of transformations to be specified in this way
- Existing tools offer hybrid implementation:
  - Object Models + State-Transition Diagrams (UML)
  - Entity Relationship Models + Data-Flow Diagrams (LBMS)
  - Entity Relationship Models + State-Transition Diagrams (*StateMate* has good STDs but - at last look - no ERDs)
- Requires
  - Intelligent Planning
  - Naming and association conventions